Abhrajyoti Kundu
Computer Science & IT (CS)

BASIC LEVEL FULL SYLLABUS TEST -1 (GATE 2023) - REPORTS

OVERALL ANALYSIS | COMPARISON REPORT | SOLUTION REPORT

ALL(65) | CORRECT(44) | INCORRECT(14) | SKIPPED(7)

Q. 21 Have any Doubt ?

Consider the following function cosmo( ):

```
int cosmo(int n)
{
    for (i1 = 1; i1 <= n; i1++)
    {
        for(i2 = 1; i2 <= i1; i2++)
        {
            for(i3 = 1; i3 <= i2; i3++)
            {
            ....
                {
                        for(im = 1; im <= im - 1; im++)
                        {
                                count++;
                        }
                }
            ....
    }
}
```

Initially, the value of the variable count is zero. Also assume the variables $i_1, i_2, \ldots i_m$ are declared before use. Then the value returned by the function call cosmo(8) when the value of m is equal to 6, will be _____.

1716 — Correct Option

Solution :
1716
The value returned will be, ${}^{n-1+m}C_m$
Putting $n = 8$, $m = 6$ we get, 1716 as the answer.

QUESTION ANALYTICS +

Q. 22 Have any Doubt ?

Which of the following statements is true?

A. An unambiguous grammar has same leftmost and rightmost derivation. — Your answer is **IN-CORRECT**

B. An LL(1) parser is a top-down parser. — Your option is **Correct**

C. LALR is more powerful than SLR. — Your option is **Correct**

D. An ambiguous grammar can never be LR(k) for any k. — Your option is **Correct**

YOUR ANSWER - a,b,c,d | CORRECT ANSWER - b,c,d | STATUS - ✗

Solution :
(b, c, d)
Except A, all the others are true. An unambiguous grammar has same leftmost and rightmost derivation parse trees, however LMDs and RMDs need not be the same.

QUESTION ANALYTICS +

Q. 23 Have any Doubt ?

Consider the following C code:

```
#include <stdio.h>
int bar(int m, int n)
{
    if(m == 0) return n;
```

```
    if(n == 0) return m;
    return bar(n% m, m);
}
int foo(int m, int n)
{
    return (m*n/bar(m, n));
}
int main( )
{
    int x = foo(1000, 1500);
    printf("%d", x);
    return 0;
}
```

The output of the program will be:

A 1000

B 2000

C 3000 — Your answer is **Correct**

Solution :
(c)
The function foo( ) computes the LCM of the integers m and n, given as input. Let's see why.
We know the relation, LCM(m, n) * GCD(m, n) = $m * n$
So we can write,

$$\text{LCM}(m, n) = m * n/\text{GCD}(m, n).$$

So we know that the LCM of 1000 and 1500 will be 3000. Therefore 3000 will be the answer.

D 4000

QUESTION ANALYTICS +

Q. 24 Have any Doubt ?

Consider the predicate $L(x, y)$: "x and y are lookalikes"
Take the domain to be the set of all people in the world. We are given two first order logic statements F_1 and F_2.

$F_1 : \exists x \exists y\ (x \neq y \wedge \text{lookalike}(x, y))$
$F_2 : \forall x \forall y \forall z\ ((x \neq y \wedge y \neq z \wedge x \neq z) \Rightarrow \neg\text{lookalike}(x, y) \vee \neg\text{lookalike}(y, z))$

Then which of the following is the correct english translation of $F_1 \wedge F_2$?

A At least 2 persons are lookalikes

B At most 2 persons are lookalikes. — Your answer is **IN-CORRECT**

C Exactly 2 persons are lookalikes — Correct Option

Solution :
(c)
F_1 : At least 2 persons are lookalikes.
F_2 : At most 2 persons are lookalikes.
Therefore, $F_1 \wedge F_2$ = Exactly 2 persons lookalikes
Therefore option (c) is correct.

D Exactly 3 persons are lookalikes

QUESTION ANALYTICS +

Q. 25 Have any Doubt ?

Consider the following C code:

```
#include<stdio.h>
int MadeEasy(int k)
{
    static int count = 0;
    while(k)
    {
        count += (k&1)?1: -1;
        k >>= 1;
    }
```

```
    return count;
}
void main( )
{
    static int x = 0;
    for(int i = 5; i > 0; i--) x = x + MadeEasy(i);
    printf("%d\n", x);
}
```

The output of the above program is _____.

8 — Correct Option

Solution :
8

Value of 'i'	Value of Count upto this point
5 (101)	1 – 1 + 1 = 1
4 (100)	1 – 2 + 1 = 0
3 (11)	0 + 1 + 1 = 2
2 (10)	2 – 1 + 1 = 2
1 (01)	2 + 1 = 3

Therefore the final value = (1 + 0 + 2 + 2 + 3) = 8

Your Answer is -8

QUESTION ANALYTICS +

Q. 26 Have any Doubt ?

The time and space complexity of the most efficient algorithm, designed to find the kth node from the end of a linked list which has n elements is equal to

A O(1), O(1)

B O(n), O(1) — Your answer is **Correct**

Solution :
(b)
There are multiple ways of getting this done in O(n) time and O(1) space.
First reverse the given linked list = O(n) time, O(1) space
Now get the k^{th} node from the beginning of the reversed linked list = O(k) time, O(1) space
So, Time complexity = O(n), Space complexity = O(1)
Thus, option (b) is correct.

C O(n), O(n)

D O(n^2), O(1)

QUESTION ANALYTICS +

Q. 27 Have any Doubt ?

Consider the following statements:

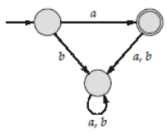
I. If a language L is finite, then the minimal DFA accepting L contains no cycle.
II. If D is a DFA having n states, and a string w is accepted by D, such that $|w| >= n$, then the DFA must contain a cycle.
III. Every DFA accepting a finite language must contain a dead state.

Which of the above statements are true?

A I, II but not III

B II, III but not I — Correct Option

Solution :
(b)
I is clearly false. Why? Consider the DFA accepting the set of strings accepting the string 'a', over $\{a, b\}$.



Clearly $L(M) = \{a\}$. Even then, we have to put a cycle to accommodate the dead state. So this fact makes I false. And it can be easily proven that for every finite language, the DFA accepting it is always going to contain a dead state. This makes III as true.
Also II is true as it follows from the pigeonhole principle. In simple terms, if a string (lets say of length n) is of the form $a_1a_2a_3 \ldots\ldots a_n$ is executed upon by a DFA, then the state sequence will be $q_0q_1q_2 \ldots\ldots q_n$ (assuming q_0 as initial state). Clearly here, our state sequence is as long as $n + 1$. But our initial assumption was that we have only n states, which means that there must a repetition in the state sequence. And, if a state is visited more than once, then it implies we've got a cycle in our DFA. Which is what the statement II claims. Therefore II is also true. So the correct choice is (b).

C Only III — Your answer is **IN-CORRECT**

D None of these

QUESTION ANALYTICS +

Q. 28 — Have any Doubt ?

Consider the following function compute( ):

```
int compute(int arr[ ], int n)
{
    int k = 0;
    for (int i = 0; i < n – 1; i++)
    for (int j = i + 1; j < n; j++)
    if (arr[i] > arr[j])
    k++;
    return (n – k);
}
```

An array A of size 16 elements is passed to this function such that for all i in $[0, 15]$, $A[i] = 2^i$. Then the output of the function compute( ) when A is passed to it will be _____.

16 — Your answer is **Correct**16

Solution :
16
If seen carefully, the program has a variable k, which is actually a counter which counts the number of inversions in the array A. So in the array A passed to the function, k will be equal to 0. Therefore $n - k$ will be equal to 16 – 0 = 16.

QUESTION ANALYTICS +

Q. 29 — Have any Doubt ?

There are 5 computers *P, Q, R, S, T* whose IP address are as follows:
P: 74.18.2.15
Q: 74.18.2.12
R: 74.18.2.14
S: 74.18.2.25
T: 74.18.2.33
The subnet mask is set to 255.255.255.248 on all computers. Then which of the following belongs to the same subnet as P?

A *Q* — Your option is **Correct**

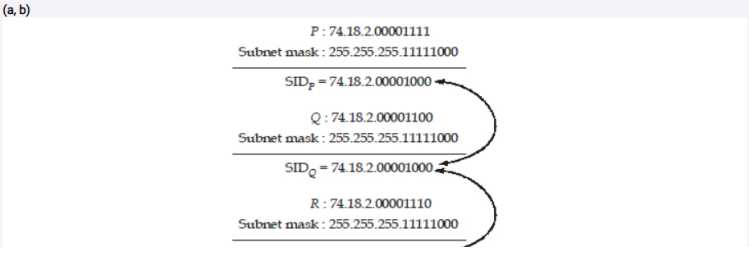
B *R* — Your option is **Correct**

C *S*

D *T*

YOUR ANSWER - a,b | CORRECT ANSWER - a,b | STATUS - ✓

Solution :
(a, b)



$SID_R = 74.18.2.00010000$

S : 74.18.2.00011001
Subnet mask : 255.255.255.11111000
$SID_S = 74.18.2.00011000$

T : 74.18.2.00100001
Subnet mask : 255.255.255.11111000
$SID_T = 74.18.2.00100000$

It can be seen that Q and R belong to the same subnet as P, therefore (a), (b) is correct.

QUESTION ANALYTICS +

Q. 30

Have any Doubt ?

Consider a DFA M. Let M' be the DFA generated by interchanging the final and non-final states of M. If M has 32 final states, and M' has 64 final states, then the total number of states in M is equal to _____.

96 Your answer is Correct96

Solution :
96

Total number of states in DFA(M) = Number of final states in M + Number of non-final states in M'
= Number of final states in M + Number of final states in M'
= 32 + 64 = 96

Therefore (96) is the answer.

QUESTION ANALYTICS +